HARSHIT SINGHANIA
2105890
CSE 9

DSA LAB ASSIGNMENT 2 LAB EXERCISES
25/07/2022

1. WAP to store n employees data such as employee name, gender, designation, department, basic pay. Calculate the gross pay of each employees as follows:

   Gross pay=basic pay + HR + DA

   HR=25% of basic, DA=75% of basic.

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

struct Employee
{
    int id, age, salary;
    char name[30], designation[30], department[30];
};

int main()
{
    struct Employee e;
    int num_records;

    printf("enter the number of records: ");
    scanf("%d", &num_records);

    for (int i = 0; i < num_records; i++)
    {
        printf("Enter the id of the Employee: ");
        scanf("%d", &e.id);

        printf("Enter the age of the Employee: ");
```

```
        scanf("%d", &e.age);

        printf("Enter the name of the Employee: ");
        getchar();
        fgets(e.name, 30, stdin);

        printf("Enter the designation of the Employee: ");
        fgets(e.designation, 30, stdin);

        printf("Enter the department of the Employee: ");
        fgets(e.department, 30, stdin);

        printf("Enter the salary of the Employee: ");
        scanf("%d", &e.salary);

        int gross_pay = e.salary + (0.25 * e.salary) + (0.75 *
e.salary);

        printf("\nEmployee Details:\n");
        printf("Employee Id: %d\n", e.id);
        printf("Employee Name: %s", e.name);
        printf("Employee age: %d\n", e.age);
        printf("Employee designation: %s", e.designation);
        printf("Employee department: %s", e.department);
        printf("Employee gross salary: %d\n", gross_pay);
    }

    return 0;
}
```

```
enter the number of records: 1
Enter the id of the Employee: 1000
Enter the age of the Employee: 34
Enter the name of the Employee: HARSHIT
Enter the designation of the Employee: MANAGER
Enter the department of the Employee: HR
Enter the salary of the Employee: 300000

Employee Details:
Employee Id: 1000
Employee Name: HARSHIT
Employee age: 34
Employee designation: MANAGER
Employee department: HR
Employee gross salary: 600000
PS C:\Users\KIIT\Desktop\Academic\3-sem\DSA(L)\LAB CODE\25-07-22>
```

2. WAP to add two distances (in km-meter) by passing structure to a function.

```c
#include <stdio.h>
#include <string.h>

struct Distance
{
    int km;
    int m;
} d1, d2, result;

void add_distance(struct Distance d1, struct Distance d2);
int main()
{
    printf("enter first distance \n");
    printf("enter km: ");
    scanf("%d", &d1.km);
    printf("enter m: ");
    scanf("%d", &d1.m);
```

```
    printf("enter second distance \n");
    printf("enter km: ");
    scanf("%d", &d2.km);
    printf("enter m: ");
    scanf("%d", &d2.m);

    add_distance(d1, d2);
    return 0;
}

void add_distance(struct Distance d1, struct Distance d2) {
    result.km = d1.km + d2.km;
    result.m = d1.m + d2.m;

    printf("the distance is %d km and %d meter", result.km,
result.m);
}
```

```
enter first distance
enter km: 1
enter m: 100
enter second distance
enter km: 2
enter m: 200
the distance is 3 km and 300 meter
PS C:\Users\KIIT\Desktop\Academic\3-sem\DSA(L)\LAB CODE\25-07-22>
```

3. Store information of 10 students using structure

```
#include <stdio.h>
struct student {
    char firstName[50];
    int roll;
    float marks;
```

```
} s[10];

int main() {
    int i;
    printf("Enter information of students:\n");

    for (i = 0; i < 10; ++i) {
        s[i].roll = i + 1;
        printf("\nFor roll number%d,\n", s[i].roll);
        printf("Enter first name: ");
        scanf("%s", s[i].firstName);
        printf("Enter marks: ");
        scanf("%f", &s[i].marks);
    }
    printf("Displaying Information:\n\n");

    for (i = 0; i < 5; ++i) {
        printf("\nRoll number: %d\n", i + 1);
        printf("First name: ");
        puts(s[i].firstName);
        printf("Marks: %.1f", s[i].marks);
        printf("\n");
    }
    return 0;
}
```

```
Enter information of students:

For roll number1,
Enter first name: HARSHIT
Enter marks: 100

For roll number2,
Enter first name: RAMESH
Enter marks: 100

For roll number3,
Enter first name: RAHUL
Enter marks: 100

For roll number4,
Enter first name: RAMAN
Enter marks: 100

For roll number5,
Enter first name: RAJESH
Enter marks: 100

For roll number6,
Enter first name: ROHAN
Enter marks: 100

For roll number7,
Enter first name: RAM
Enter marks: 100

For roll number8,
Enter first name: RISHI
Enter marks: 100
```

4. Add two complex numbers by passing structures to a function

```c
#include <stdio.h>

struct complex
{
    float real;
    float imaginary;
} c1, c2, add, result;

void sum(struct complex c1, struct complex c2) {
    struct add;
    add.real = c1.real + c2.real;
    add.imaginary = c1.imaginary + c2.imaginary;
    printf("SUM = %0.2f + i %0.2f", add.real, add.imaginary);
}

int main()
{

    struct complex cnum1, cnum2;
    printf("Enter real and imaginary part of first complex
number:\n");
    scanf("%f%f", &cnum1.real, &cnum1.imaginary);
    printf("Enter real and imaginary part of second complex
number:\n");
    scanf("%f%f", &cnum2.real, &cnum2.imaginary);
    sum(cnum1, cnum2);
    return 0;
}
```

```
Enter real and imaginary part of first complex number:
1 2
Enter real and imaginary part of second complex number:
2
1
SUM = 3.00 + i 3.00
PS C:\Users\KIIT\Desktop\Academic\3-sem\DSA(L)\LAB CODE\25-07-22>
```

5. Calculate the difference between two time periods

```c
#include<stdio.h>

struct time
{
    int hours;
    int minutes;
    int seconds;
};

int main()
{
    struct time start, stop, diff;

    printf("Enter hours, minutes and seconds of start time: ");
    scanf("%d%d%d", &start.hours,&start.minutes,
&start.seconds);
    printf("Enter hours, minutes and seconds of stop time: ");
    scanf("%d%d%d", &stop.hours,&stop.minutes, &stop.seconds);


    diff.seconds = stop.seconds - start.seconds;
    diff.minutes = stop.minutes - start.minutes;
    diff.hours = stop.hours - start.hours;

    printf("Difference = %d : %d : %d", diff.hours,
diff.minutes, diff.seconds);
```

```
    return 0;
}
```

```
Enter hours, minutes and seconds of start time: 1 12 13
Enter hours, minutes and seconds of stop time: 4 14 18
Difference = 3 : 2 : 5
PS C:\Users\KIIT\Desktop\Academic\3-sem\DSA(L)\LAB CODE\25-07-22>
```

6. Store information of n students using structures and Dynamic Memory Allocation.

```
#include <stdio.h>
#include <stdlib.h>
struct course
{
    int marks;
    char subject[30];
};

int main()
{
    struct course *ptr;
    int noOfRecords;
    printf("Enter the number of records: ");
    scanf("%d", &noOfRecords);

    // Memory allocation for noOfRecords structures
    ptr = (struct course *)malloc(noOfRecords * sizeof(struct
course));
    for (int i = 0; i < noOfRecords; ++i)
    {
        printf("Enter subject and marks:\n");
        scanf("%s %d", (ptr + i)->subject, &(ptr + i)->marks);
    }

    printf("Displaying Information:\n");
```

```
    for (int i = 0; i < noOfRecords; ++i)
    {
        printf("%s\t%d\n", (ptr + i)->subject, (ptr +
i)->marks);
    }

    free(ptr);

    return 0;
}
```

```
Enter the number of records: 1
Enter subject and marks:
maths 33
Displaying Information:
maths    33
```

7. C program to read a one dimensional array, print sum of all elements along with inputted array elements using Dynamic Memory Allocation.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int n, i, *ptr, sum = 0;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    ptr = (int *)malloc(n * sizeof(int));

    if (ptr == NULL)
    {
```

```
        printf("Error! memory not allocated.");
        exit(0);
    }

    printf("Enter elements: ");
    for (i = 0; i < n; ++i)
    {
        scanf("%d", ptr + i);
        sum += *(ptr + i);
    }

    for (i = 0; i < n; ++i)
    {
        printf("%d ", ptr[i]);
    }

    printf("Sum = %d", sum);

    free(ptr);

    return 0;
}
```

```
Enter number of elements: 5
Enter elements: 1
2
3
4
5
1 2 3 4 5 Sum = 15
PS C:\Users\KIIT\Desktop\Academic\3-sem\DSA(L)\LAB CODE\25-07-22>
```

DSA LAB ASSIGNMENT 2 HOME EXERCISES

1. WAP to print all permutations of a given string using pointers.

```c
#include <stdio.h>
#include <string.h>

void change_pos(char *ch1, char *ch2)
{
    char temp;
    temp = *ch1;
    *ch1 = *ch2;
    *ch2 = temp;
}

void change_permu(char *cht, int stno, int endno)
{
    int i;
    if (stno == endno)
        printf("%s\n", cht);
    else
        for (i = stno; i <= endno; i++)
        {
            change_pos(cht + stno, cht + i);
            change_permu(cht, stno + 1, endno);
            change_pos(cht + stno, cht + i);
        }
}

int main()
{
    char str[] = "abc";
    int n = strlen(str);
    printf("the permutations of %s are:", str);
```

```
    change_permu(str, 0, n - 1);
    printf("\n\n");
    return 0;
}
```

```
the permutations of abc are:abc
acb
bac
bca
cba
cab
```

2. WAP to replace every array element by multiplication of previous and next of an n element.

```
#include <stdio.h>
void newArryPrevNext(int arr1[], int n)
{
    if (n <= 1)
        return;
    int pre_elem = arr1[0];
    arr1[0] = arr1[0] * arr1[1];
    for (int i = 1; i < n - 1; i++)
    {
        int cur_elem = arr1[i];
        arr1[i] = pre_elem * arr1[i + 1];
        pre_elem = cur_elem;
    }
    arr1[n - 1] = pre_elem * arr1[n - 1];
}
int main()
{
    int arr1[] = {1, 2, 3, 4, 5, 6};
    int n = sizeof(arr1) / sizeof(arr1[0]);
```

```
    int i = 0;

    printf("The given array is:  \n");
    for (i = 0; i < n; i++)
    {
        printf("%d  ", arr1[i]);
    }
    printf("\n");

    printf("The new array is: \n");
    newArryPrevNext(arr1, n);
    for (int i = 0; i < n; i++)
        printf("%d ", arr1[i]);
    return 0;
}
```

```
The given array is:
1  2  3  4  5  6
The new array is:
2 3 8 15 24 30
PS C:\Users\KIIT\Desktop\Academic\3-sem\DSA(L)\18-07-22>
```

3. WAP to arrange the elements of an array such that all even numbers are followed by all odd numbers.

```
#include <stdio.h>
#include <conio.h>

int main()
{
    int a[10000], b[10000], i, n, j, k, temp, c = 0;

    printf("Enter size of the  array : ");
```

```
    scanf("%d", &n);
    printf("Enter elements in array : ");
    for (i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
        if (a[i] % 2 == 1)
            c++;
    }
    for (i = 0; i < n - 1; i++)
    {

        for (j = 0; j < n - i - 1; j++)
        {
            if (a[j] > a[j + 1])
            {
                temp = a[j];
                a[j] = a[j + 1];
                a[j + 1] = temp;
            }
        }
    }

    k = 0;
    j = n - c;

    for (i = 0; i < n; i++)
    {
        if (a[i] % 2 == 0)
        {
            if (k < n - c)
                b[k++] = a[i];
        }
        else
        {
```

```
            if (j < n)
                b[j++] = a[i];
        }
    }

    printf("\narray after sorting even and odd elements
separately:\n ");

    for (i = 0; i < n; i++)
    {
        a[i] = b[i];
        printf("%d ", a[i]);
    }
}
```

```
Enter size of the  array : 10
Enter elements in array : 1
2
3
4
5
6
7
8
9
9
10

array after sorting even and odd elements separately:
 2 4 6 8 10 1 3 5 7 9
PS C:\Users\KIIT\Desktop\Academic\3-sem\DSA(L)\LAB CODE\18-07-22>
```

4. WAP to find the largest number and counts the occurrence of the largest number in an array of n integers using a single loop.

```c
#include <stdio.h>

void maxoccurence(int a[], int ar_size)
{
    int max = a[0], count = 0, i;

    for (i = 0; i < ar_size; i++)
    {
        if (a[i] == max)

            count++;

        if (a[i] > max)
        {
            max = a[i];

            count = 1;
        }
    }

    printf("Maximum element in the array is %d\n", max);
    printf("the maximum occurence of %d is %d\n", max, count);
}

int main()
{
    int n, i, a[100];
    printf("Enter the number of elements in the array: ");
    scanf("%d", &n);
    for (i = 0; i < n; i++)
    {
```

```
        printf("Enter the element %d: ", i + 1);
        scanf("%d", &a[i]);
    }
    maxoccurence(a, n);
}
```

```
Enter the number of elements in the array: 10
Enter the element 1: 1
Enter the element 2: 2
Enter the element 3: 3
Enter the element 4: 4
Enter the element 5: 5
Enter the element 6: 6
Enter the element 7: 7
Enter the element 8: 8
Enter the element 9: 9
Enter the element 10: 9
Maximum element in the array is 9
the maximum occurence of 9 is 2
```